

CLAIMS

What is claimed is:

1. A method in a data processing system for developing a data flow program comprising code segments distributed between blocks of memory, the method comprising the steps of:

generating a graph that represents the data flow program, the graph comprising nodes corresponding to selected ones of the blocks and arcs corresponding to dependency relationships between the nodes; and

receiving an optimization command to manipulate the graph to improve performance of the data flow program.

2. The method of claim 1, further comprising the step of performing the optimization command.

3. The method of claim 2, further comprising the step of performing performance analysis on the data flow program in accordance with the optimization command.

4. A method according to claim 1, wherein the nodes are placed in an execution queue for processing, and wherein the optimization command specifies re-ordering the nodes in the execution queue.

5. A method according to claim 1, wherein the nodes are characterized by node execution times, and wherein the optimization command specifies setting one of the node execution times, and further comprising the step of simulating execution of the data flow program in accordance with the node execution times.

6. A method according to claim 1, wherein the blocks are assigned data operated on by the data flow program and wherein the optimization command specifies setting revised data for a selected block.

7. A method according to claim 1, wherein the optimization command specifies a performance comparison between selected nodes.

6. A method in a data processing system for developing a data flow program comprising code segments that operate on data in memory, the method comprising the steps of:

dividing a memory area into blocks and associating each block with at least a portion of the data and with at least one code segment;

generating a graph representation of the data flow program, the graph representation comprising nodes associated with the blocks, and arcs associated with dependencies between the blocks; and

performing an optimization command to manipulate the graph to improve performance of the data flow program.

7. The method of claim 6, further comprising the step of performing performance analysis on the data flow program.

8. The method of claim 6, further comprising the step of entering the nodes in a queue for execution, and wherein the optimization command specifies reordering the nodes in the queue.

9. The method of claim 7, wherein the step of performing performance analysis comprises the step of determining execution time for the data flow program.

10. The method of claim 7, wherein the step of performing performance analysis comprises the step of simulating execution of the nodes in the graph.

11. The method of claim 7, wherein the nodes are characterized by node execution times, and wherein the optimization command specifies a reduced node execution time for one of the nodes, and wherein the step of performing performance analysis comprises the step of determining execution time for the data flow program in accordance with the reduced node execution time.

12. The method of claim 7, wherein the optimization command specifies a memory bandwidth, and wherein the step of performing performance analysis comprises the step of determining execution time for the data flow program in accordance with the memory bandwidth.

13. The method of claim 6, wherein the optimization command specifies a modification to at least a portion of the data.

14. A computer-readable medium containing instructions that cause a data processing system to perform a method for developing a data flow program comprising code segments that operate on data in memory, the method comprising the steps of:

dividing the memory into blocks and associating each block with at least a portion of the data and with at least one code segment;

generating a graph representation of the data flow program, the graph representation comprising nodes associated with the blocks, and arcs associated with dependencies between the blocks; and

performing an optimization command to manipulate the graph to improve performance of the data flow program.

15. The computer-readable medium of claim 14, further comprising the step of performing performance analysis on the data flow program.
16. The computer-readable medium of claim 14, further comprising the step of entering the nodes in a queue for execution, and wherein the optimization command specifies reordering the nodes in the queue.
17. The computer-readable medium of claim 15, wherein the step of performing performance analysis comprises the step of determining execution time for the data flow program.
18. The computer-readable medium of claim 15, wherein the step of performing performance analysis comprises the step of simulating execution of the nodes in the graph.
19. The computer-readable medium of claim 15, wherein the nodes are characterized by node execution times, and wherein the optimization command specifies a reduced node execution time for one of the nodes, and wherein the step of performing performance analysis comprises the step of determining execution time for the data flow program in accordance with the reduced node execution time.
20. The computer-readable medium of claim 14, wherein the optimization command specifies a modification to at least a portion of the data.
21. A data processing system comprising:
 - a memory comprising a data flow program and a data flow development tool that generates a graph representation of the data flow program by associating data processed by the data flow program to blocks in the memory, by associating code segments of the data flow program to the blocks, and by

determining dependencies between the blocks, that executes code segments in parallel using multiple threads, and that performs an optimization command to manipulate the graph to improve performance of the data flow program; and

a processor that runs the data flow development tool.

22. The data processing system of claim 21, further comprising a queue in memory in which the blocks are placed for execution, and wherein the optimization comprises reordering the blocks in the queue.

23. The data processing system of claim 21, wherein the optimization command comprises an execution time specification of for one of the blocks.

24. The data processing system of claim 21, wherein the optimization command comprises a block performance comparison command.

25. The data processing system of claim 21, wherein the optimization command comprises a block data modification command.

26. A data processing system for developing a data flow program comprising code segments that operate on data in memory, the data processing system comprising:

means for apportioning a memory into regions and associating the data and the code segments of the data flow program with the regions;

means for determining dependencies between the regions;

means for generating a graph representation of the data flow program, the representation comprising nodes associated with the blocks, and dependencies between blocks; and

means for obtaining an optimization command specifying an optimization on the data flow program.